UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/828,794	04/21/2004	Erik Altman	YOR090040015US1 (163-29)	5081

24336 7590 09/20/2006

KEUSEY, TUTUNJIAN & BITETTO, P.C.
20 CROSSWAYS PARK NORTH
SUITE 210
WOODBURY, NY 11797

EXAMINER	
FENNEMA, ROBERT E	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 09/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/828,794	ALTMAN ET AL.	
	Examiner	Art Unit	
	Robert E. Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 August 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-30 have been considered. Claims 1, 13, and 22 have been amended as per Applicant's request.

Claim Rejections - 35 USC § 102

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-30 are rejected under 35 U.S.C. 102(b) as being anticipated by Tran (USPN 5,765,035).

3. As per Claim 1, Tran teaches: A pipeline, comprising:

a plurality of operational stages (Abstract), the stages including:

a pointer register stage which stores pointer information and updates (Column 9, Lines 32-37, the reservation stations. The stations hold instruction information to be executed by the functional units. Column 7, Lines 31-57 disclose that registers, which generally hold the information in the reservation stations, can make use of indirect addressing, such that the value of the register is used as an address (or pointer) to a location in memory, where the real data is found. In this embodiment, the basic x86 registers function as "pointers", which can be stored in the reservation station prior to execution);

a dependency checking stage located downstream of the pointer register stage, which determines if instruction dependencies exist and stalls an issue prior to issuance if necessary to resolve inter-instruction dependencies (Column 7, Lines 60-64, the

reorder buffer does dependency checking, and as seen by Column 9, Lines 51-64, the reorder buffer "tags" an entry in a reservation station if an operand is not available, causing the instruction to stall, being unable to issue until the tag is replaced by a value. Column 11, Line 54 – Column 12, Line 17 deals with stalling and dependency checking in the memory operation case);

at least one functional unit providing pointer information updates to the pointer register stage (Column 9, Lines 45-50).

4. As per Claim 2, Tran teaches: The pipeline as recited in claim 1, further comprising a pointer execution stage used before the dependency checking stage such that inter-instruction dependency is checked after the pointer execution stage or in parallel with pointer execution (Column 9, Lines 45-50. The functional units are part of the pointer execution stage).

5. As per Claim 3, Tran teaches: The pipeline as recited in claim 2, further comprising a path for making pointer updates available to the pointer register stage before the instruction reaches a write back stage of the pipeline (Column 9, Lines 45-50. The result is bypassed back to the reservation stations (pointer register stage), as the same time it goes to the reorder buffer).

6. As per Claim 4, Tran teaches: The pipeline as recited in claim 3, wherein the path includes a normal pointer update path which returns pointer information from the at

least one functional unit (Column 9, Lines 45-50).

7. As per Claim 5, Tran teaches: The pipeline as recited in claim 3, further comprising a pointer execution bypass making pointer updates available to immediately following instructions before a pointer update is written into the pointer register file (Column 9, Lines 45-50).

8. As per Claim 6, Tran teaches: The pipeline as recited in claim 2, further comprising a pointer reorder buffer coupled to the pointer register stage to maintain a precise state of pointers (Column 7, Lines 60-64, it is part of the dependency checking stage).

9. As per Claim 7, Tran teaches: The pipeline as recited in claim 6, further comprising a precise pointer file for storing the precise state of the pointer reorder buffer (Figure 1, Register File 218).

10. As per Claim 8, Tran teaches: The pipeline as recited in claim 7, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file (Column 10, Lines 6-16).

11. As per Claim 9, Tran teaches: The pipeline as recited in claim 1, further comprising a combined pointer reorder buffer/issue stage coupled to the dependence checking stage to issue instructions and maintain a precise state/order of pointers (Column 7, Lines 60-64, and Column 8, Line 46 – Column 9, Line 9. The reorder buffer maintains the ordering of the operations, and helps issue in the sense that it forwards appropriate data to the reservation station to allow it to issue).

12. As per Claim 10, Tran teaches: The pipeline as recited in claim 9, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers (Column 8, Line 46 – Column 9, Line 6).

13. As per Claim 11, Tran teaches: The pipeline as recited in claim 9, further comprising a precise pointer file for storing the precise state of the combined pointer reorder buffer/issue stage (Figure 1, Register File 218).

14. As per Claim 12, Tran teaches: The pipeline as recited in claim 11, further comprising an interrupt recovery path, which restores the pointer register stage to the precise state from the precise pointer file (Column 10, Lines 6-16).

15. As per Claim 13, Tran teaches: A pipeline, comprising:

a plurality of operational stages (Abstract), the stages including:

a pointer register stage which stores pointer information and updates (Column 9, Lines 32-37, the reservation stations. The stations hold instruction information to be executed by the functional units. Column 7, Lines 31-57 disclose that registers, which generally hold the information in the reservation stations, can make use of indirect addressing, such that the value of the register is used as an address (or pointer) to a location in memory, where the real data is found. In this embodiment, the basic x86 registers function as "pointers", which can be stored in the reservation station prior to execution);

a dependence checking stage located downstream of the pointer register stage, which determines if instruction dependencies exist and stalls an issue prior to issuance if necessary to resolve inter-instruction dependencies (Column 7, Lines 60-64, the reorder buffer does dependency checking, and as seen by Column 9, Lines 51-64, the reorder buffer "tags" an entry in a reservation station if an operand is not available, causing the instruction to stall, being unable to issue until the tag is replaced by a value. Column 11, Line 54 – Column 12, Line 17 deals with stalling and dependency checking in the memory operation case);

a pointer execution stage for processing pointers prior to the dependence checking stage (Column 9, Lines 45-50. The functional units are part of the pointer execution stage), the pointer execution stage providing pointer updates to the pointer register stage via an early pointer update path (Column 9, Lines 45-50. The result is bypassed back to the reservation stations (pointer register stage), as the same time it goes to the reorder buffer); and

at least one functional unit providing pointer information updates to the pointer register stage such that pointer information is processed and updated to the pointer register stage (Column 9, Lines 45-50).

16. As per Claim 14, Tran teaches: The pipeline as recited in claim 13, further comprising a pointer reorder buffer coupled to the pointer register stage to maintain a precise state of pointers (Column 7, Lines 60-64, it is part of the dependency checking stage).

17. As per Claim 15, Tran teaches: The pipeline as recited in claim 14, further comprising a precise pointer file for storing the precise state of the pointer reorder buffer (Figure 1, Register File 218).

18. As per Claim 16, Tran teaches: The pipeline as recited in claim 15, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file (Column 10, Lines 6-16).

19. As per Claim 17, Tran teaches: The pipeline as recited in claim 13, further comprising a normal pointer update path which returns pointer information from the at least one functional unit (Column 9, Lines 45-50).

20. As per Claim 18, Tran teaches: The pipeline as recited in claim 13, further comprising a combined pointer reorder buffer/issue stage coupled to the dependence checking stage to issue instructions and maintain a precise state/order of pointers (Column 7, Lines 60-64, and Column 8, Line 46 – Column 9, Line 9. The reorder buffer maintains the ordering of the operations, and helps issue in the sense that it forwards appropriate data to the reservation station to allow it to issue).

21. As per Claim 19, Tran teaches: The pipeline as recited in claim 18, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers (Column 8, Line 46 – Column 9, Line 6).

22. As per Claim 20, Tran teaches: The pipeline as recited in claim 18, further comprising a precise pointer file for storing the precise state of the combined pointer reorder buffer/issue stage (Figure 1, Register File 218).

23. As per Claim 21, Tran teaches: The pipeline as recited in claim 20, further comprising an interrupt recovery path, which restores the pointer register stage to the precise state from the precise pointer file (Column 10, Lines 6-16).

24. As per Claim 22, Tran teaches: A method for updating pointers ahead of an instruction, comprising the steps of:

providing a plurality of operational stages (Abstract), including a pointer register stage which stores pointer information and updates (Column 9, Lines 32-37, the reservation stations. The stations hold instruction information to be executed by the functional units. Column 7, Lines 31-57 disclose that registers, which generally hold the information in the reservation stations, can make use of indirect addressing, such that the value of the register is used as an address (or pointer) to a location in memory, where the real data is found. In this embodiment, the basic x86 registers function as "pointers", which can be stored in the reservation station prior to execution), a dependence checking stage located downstream of the pointer register stage, which determines if instruction dependencies exist and stalls an issue prior to issuance if necessary to resolve inter-instruction dependencies (Column 7, Lines 60-64, the reorder buffer does dependency checking, and as seen by Column 9, Lines 51-64, the reorder buffer "tags" an entry in a reservation station if an operand is not available, causing the instruction to stall, being unable to issue until the tag is replaced by a value. Column 11, Line 54 – Column 12, Line 17 deals with stalling and dependency checking in the memory operation case), and at least one functional unit providing pointer information updates to the pointer register stage (Column 9, Lines 45-50); and

processing pointer information to update the pointer information for the pointer register stage so that updated pointer information is available (Column 8, Lines 60-65).

25. As per Claim 23, Tran teaches: The method as recited in claim 22, further comprising a step of providing pointer updates to the pointer register stage via an early

pointer update path by providing a pointer execution stage used before the pointer register stage and the dependence checking stage (Column 9, Lines 45-50. The result is bypassed back to the reservation stations (pointer register stage), as the same time it goes to the reorder buffer).

26. As per Claim 24, Tran teaches: The method as recited in claim 23, further comprising a step of maintaining a precise state of pointers by employing a pointer reorder buffer (Column 7, Lines 60-64, it is part of the dependency checking stage).

27. As per Claim 25, Tran teaches: The method as recited in claim 24, further comprising a step of storing the precise state of the pointer reorder buffer in a precise pointer file (Figure 1, Register File 218 holds the precise state).

28. As per Claim 26, Tran teaches: The method as recited in claim 24, further comprising updating reordering or recovery information from the precise pointer file using an interrupt recovery path to the pointer register stage (Column 10, Lines 6-16).

29. As per Claim 27, Tran teaches: The method as recited in claim 23, further comprising maintaining a precise state/order of pointers using a combined pointer reorder buffer/issue stage coupled to the rename and dependence checking stage (Column 7, Lines 60-64, and Column 8, Line 46 – Column 9, Line 9. The reorder buffer maintains the ordering of the operations, and helps issue in the sense that it forwards

appropriate data to the reservation station to allow it to issue).

30. As per Claim 28, Tran teaches: The method as recited in claim 27, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers (Column 8, Line 46 – Column 9, Line 6).

31. As per Claim 29, Tran teaches: The method as recited in claim 27, further comprising storing the precise state of the combined pointer reorder buffer/issue stage using a precise pointer file (Figure 1, Register File 218).

32. As per Claim 30, Tran teaches: The method as recited in claim 29, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file (Column 10, Lines 6-16).

Response to Arguments

33. Applicant's arguments filed 8/7/2006 have been fully considered but they are not persuasive. Applicant has argued that Tran fails to disclose or suggest a dependency checking stage that is downstream of a pointer register stage, which stalls an issue prior to issuance, and instead states that Tran's reorder buffer is located downstream of the pointer register stage, but that it stalls after issuance. Tran does stall a memory

operation after it has left a reservation station, however, referring to Column 9, Lines 40-64, the Reservation station holds instructions waiting to issue. They can only be issued when all available operands are available, as it is not possible to execute an instruction without available data to operate on. If the data is not available, the reorder buffer will send a tag to the reservation station indicating the data is not available, and designating where the data will come from. This effectively stalls the issue, prior to issuance. As in stated in Column 9, Lines 62-64, the Reorder buffer does this to ensure data coherency. Thus, Tran stalls an issue prior to issuance, for non-memory operations to be executed by a functional unit.

However, the term "issuance" could also be read to not be when an instruction leaves the reservation station, but when an instruction is "issued" to its executing unit, whether it be the functional units, or the memory, IE, when it is time to perform the operation specified by its instruction. In the latter interpretation, Tran does stall an issue "prior to issuance" if necessary for memory operations as well, because as disclosed in both Column 9, Lines 40-64, and Column 11, Line 54 – Column 12, Line 18, an instruction can not be executed if dependant on something else that had not been completed, so all instructions must necessarily be stalled prior to being executed if they can not execute in order to ensure correct processor operation. So in this interpretation of "issue", Tran does stall a memory instruction prior to issuance, however, a memory operation would have a different "issuance" time than a functional operation. Not given a concrete definition of "issue" in the claims, Examiner is taking the broadest reasonable

interpretation of "issue", and has interpreted issuance in this manner. However, Examiner has amended the claim rejections above to deal with both interpretations.

Conclusion

34. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:00-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert E Fennema
Examiner
Art Unit 2183

RF

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100